

STUDY OF SPEAKER RECOGNITION SYSTEMS

A THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR

BACHELOR IN TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION

BY

ASHISH KUMAR PANDA (107EC016)

AMIT KUMAR SAHOO (107EC014)

UNDER THE GUIDANCE

OF

PROF. SUKADEV MEHER



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

2007-2011

CERTIFICATE



NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

This is to certify that the thesis titled, “Study of Speaker Recognition Systems” submitted by Ashish Kumar Panda (107EC016) and Amit Kumar Sahoo (107EC014) in partial fulfilments for the requirements for the award of Bachelor of Technology Degree in Electronics and Communication Engineering, National Institute of Technology, Rourkela is an authentic work carried out by them is under my supervision.

Date:

Prof. Sukadev Meher

Department of Electronics and Communication
Engineering

ACKNOWLEDGEMENT

First of all we would like to express our deep gratitude towards our advisor and guide **Prof. Sukadev Meher** who has always been a guiding force behind this project work. His highly influential personality has provided us constant encouragement to tackle any difficult task assigned. We are indebted to him for his invaluable advice and for propelling us further in every aspect of our academic life. His depths of knowledge, crystal clear concepts have made our academic journey a cake walk. We consider it our good fortune to have got an opportunity to work with such a wonderful personality.

Next, we want to express our respect to **Prof. Samit Ari, Prof. S.K. Patra** and **Prof. A.K Sahoo** for providing us necessary information about project work and helping us learn various tough concepts. They have always been great sources of inspiration to us and we would like to convey our deep regards to them. We are grateful to all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their generous help in various ways for the completion of this thesis.

We would also like to highlight the names of **Ajay** and **Ashish** for helping us a lot during the thesis period. We would like to thank all our friends and especially our classmates for all the thought provoking discussions we had, which inspired us to think beyond the obvious. We've enjoyed their companionship a lot during our stay at NIT Rourkela.

We are especially indebted to our parents for their love, sacrifice, and support. They are our first teachers after we came to this world and have always been mile stones to lead us a disciplined life.

Ashish Kumar Panda (107EC016)
Dept of ECE, NIT, Rourkela

Amit Kumar Sahoo (107EC014)
Dept of ECE, NIT, Rourkela

ABSTRACT

Speaker Recognition is the computing task of validating a user's claimed identity using characteristics extracted from their voices. This technique is one of the most useful and popular biometric recognition techniques in the world especially related to areas in which security is a major concern. It can be used for authentication, surveillance, forensic speaker recognition and a number of related activities.

Speaker recognition can be classified into identification and verification. Speaker identification is the process of determining which registered speaker provides a given utterance. Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker.

The process of Speaker recognition consists of 2 modules namely: - feature extraction and feature matching. Feature extraction is the process in which we extract a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves identification of the unknown speaker by comparing the extracted features from his/her voice input with the ones from a set of known speakers.

Our proposed work consists of truncating a recorded voice signal, framing it, passing it through a window function, calculating the Short Term FFT, extracting its features and matching it with a stored template. Cepstral Coefficient Calculation and Mel frequency Cepstral Coefficients (MFCC) are applied for feature extraction purpose. VQLBG (Vector Quantization via Linde-Buzo-Gray), DTW (Dynamic Time Warping) and GMM (Gaussian Mixture Modelling) algorithms are used for generating template and feature matching purpose.

CONTENTS

	Page no.
Certificate	2
Acknowledgement	3
Abstract	4
List of Figures	8
CHAPTER 1 INTRODUCTION	9
1.1 INTRODUCTION	10
1.2 BIOMETRICS	11
1.3 BIOMETRIC SYSTEM	12
1.4 PREVIOUS WORK	13
1.5 THESIS CONTRIBUTION	14
1.6 OUTLINE OF THESIS	15
CHAPTER 2 PRINCIPLES OF SPEAKER RECOGNITION	16
2.1 INTRODUCTION	17
2.2 CLASSIFICATION	18
2.2.1 Open Set vs Closed Set	18
2.2.2 Identification vs Verification	19
2.2.3 Text-Dependent vs Text-Independent	21
2.3 MODULES	21
2.4 INITIALS	22
2.5 SUMMARY	22
CHAPTER 3 SPEECH FEATURE EXTRACTION	23
3.1 INTRODUCTION	24
3.2 PRE- PROCESSING	25
3.2.1 Truncation	25
3.2.2 Frame Blocking	26
3.2.3 Windowing	27
3.2.4 Short Term Fourier Transform	28

3.3 CEPSTRAL COEFFICIENTS USING DCT	28
3.4 MFCC (MEL FREQUENCY CEPSTRAL COEFFICIENTS)	29
3.4.1 Mel-Frequency Wrapping	29
3.4.2 Cepstrum	31
3.5 SUMMARY	31
 CHAPTER 4 FEATURE MATCHING	 32
4.1 INTRODUCTION	33
4.2 SPEAKER MODELING	34
4.3 VECTOR QUANTIZATION	35
4.4 OPTIMIZATION USING LBG ALGORITHM	39
4.5 DTW (DYNAMIC TIME WARPING) ALGORITHM	41
4.5.1 Introduction	41
4.5.2 Classical DTW Algorithm	42
4.5.3 Dynamic Programming	43
4.5.4 Speaker Identification	44
4.6 GMM (GAUSSIAN MIXTURE MODELING)	45
4.6.1 Introduction	45
4.6.2 Model Description	46
4.6.3 Maximum Likelihood Parameter Estimation	47
4.6.4 Speaker Identification	48
4.7 SUMMARY	49
 CHAPTER 5 RESULTS	 50
5.1 OVERVIEW	51
5.2 FEATURE EXTRACTION	51
5.2.1 Cepstral Coefficients	51
5.2.2 MFCC	53
5.3 FEATURE MATCHING	54
5.3.1 VQ using LBG algorithm	54
5.3.2 DTW (Dynamic Time Warping) Algorithm	55
5.3.3 GMM (Gaussian Mixture Modeling)	56

CHAPTER 6 CONCLUSION	57
6.1 CONCLUSION	58
REFERENCES	59

LIST OF FIGURES

Figure 1.1 Basic Block Diagram of a Biometric System	12
Figure 2.1 Classification of Speaker Recognition	18
Figure 2.2 Block Diagrams of Identification and Verification systems	19
Figure 2.3 Practical examples of Identification and Verification Systems	20
Figure 2.4 Modules of a speaker recognition system	21
Figure 3.1 Example of a Speech Signal	24
Figure 3.2 Truncated version of original signal	25
Figure 3.3 Frame output of truncated signal	26
Figure 3.4 Hamming Window	27
Figure 3.5 MFCC Processor	29
Figure 3.6 Example of a Mel-spaced frequency bank	30
Figure 4.1 Codewords in 2-dimensional space	37
Figure 4.2 Block Diagram of the basic VQ Training and classification structure	37
Figure 4.3 Conceptual diagram illustrating vector quantization codebook formation	38
Figure 4.4 Flow diagram of the LBG algorithm	40
Figure 4.5 Dynamic Time Warping	41
Figure 4.6 Constellation diagram	43
Figure 4.7: Cumulative matrix of two time series data	44
Figure 4.8: GMM model showing a feature space and corresponding Gaussian model	45
Figure 4.9: Description of M-component Gaussian densities	47
Figure 5.1 Result of Cepstral Coefficient Calculation	52
Figure 5.2: Feature vectors and MFCC	53
Figure 5.3: VQLBG output and corresponding VQ distortion matrix	54
Figure 5.4: Speaker recognition using DTW	55
Figure 5.5: Output of GMM	56

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Speaker recognition is the process of recognizing automatically who is speaking on the basis of individual information included in speech waves. This technique uses the speaker's voice to verify their identity and provides control access to services such as voice dialing, database access services, information services, voice mail, security control for confidential information areas, remote access to computers and several other fields where security is the main area of concern.

Speech is a complicated signal produced as a result of several transformations occurring at several different levels: semantic, linguistic, articulatory, and acoustic. Differences in these transformations are reflected in the differences in the acoustic properties of the speech signal. Besides there are speaker related differences which are a result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits of different individuals. In speaker recognition, all these differences are taken into account and used to discriminate between speakers [10].

The forthcoming chapters describe how to build a simple and representative automatic speaker recognition system. Such a speaker recognition system helps in the basic purpose of speaker identification which forms a formidable domain in the field of speaker recognition. The system designed has potential in several security applications. Examples may include, users having to speak a PIN (Personal Identification Number) in order to gain access to the laboratory they work in, or having to speak their credit card number over the telephone line to verify their identity. By checking the voice characteristics of the input utterance, using an automatic speaker recognition system similar to the one that we will describe, the system is able to add an extra level of security.

1.2 BIOMETRICS

A biometric system is a pattern recognition system, which makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristics possessed by the user. It comprises methods for uniquely recognising humans based upon one or more intrinsic physical or behavioral traits. Biometrics is used as a form of identity access management and access control. It is also used to identify individuals in groups that are under surveillance. Biometric characteristics can be divided into 2 main classes:-

- **Physiological**

It is related to the shape of the body. Examples include

1. Fingerprint recognition
2. Face recognition
3. D.N.A
4. Palm print
5. Hand geometry
6. Iris recognition

- **Behavioral**

It is related to behaviour of the person, Examples include typing

1. Rhythm
2. Gait
3. Voice

Strictly speaking, voice is also a physiological trait because every person has a different vocal tract, but voice (speaker) recognition is mainly based on the study of the way a person speaks, commonly classified as behavioral. Among the above, the most popular biometric system is the speaker (voice) recognition system because of its easy implementation and economical hardware [18].

1.3 BIOMETRIC SYSTEM

This section provides the basic structure of a biometric system. The first time an individual uses a biometric system is called an enrollment. During the enrollment, biometric information from the individual is stored. In subsequent uses, biometric information is detected and compared with the information stored at the time of enrollment. The first block (sensor) is the interface between the real world and the system i.e. it has to acquire all the necessary data from the real world. The second block performs all the necessary pre-processing i.e. it has to remove the artifacts from the sensor to enhance the input (e.g. removing background noise), to use some kind of normalization etc. In the third block necessary features are extracted. A vector of numbers or an image with particular properties is used to create a template. A template is a synthesis of the relevant characteristics extracted from the source. Elements of the biometric measurement that are not used in the comparison algorithm are discarded in the template to reduce the file size and to protect the identity of the enrollee [18]. Figure 1.1 below gives the basic block diagram of a biometric system:-

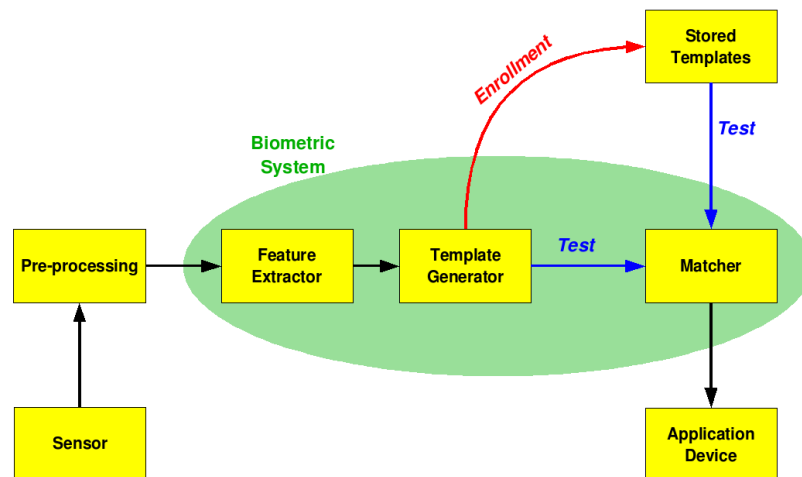


Figure 1.1: Basic Block Diagram of a Biometric System

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. Similar to a biometric system, it has two sessions:-

- **Enrollment session or Training phase**

In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference model for that speaker.

- **Operation session or Testing phase**

During the testing (operational) phase, the input speech is matched with stored reference model(s) and recognition decision is made.

1.4 PREVIOUS WORK

A considerable number of speaker-recognition activities are being carried out in industries, national laboratories and universities. Several enterprises and universities have carried out intense research activities in this domain and have come up with various generations of speaker-recognition systems. Those institutions include AT&T and its derivatives (Bolt, Beranek, and Newman) [4]; the Dalle Molle Institute for Perceptual Artificial Intelligence (Switzerland); MIT Lincoln Labs; National Tsing Hua University (Taiwan); Nippon Telegraph and Telephone (Japan); Rutgers University and Texas Instruments (TI) [1]. Sandia National Laboratories, National Institute of Standards and Technology, the National Security Agency etc. have conducted evaluations of speaker-recognition systems. It is to be noted that it is difficult to make reasonable comparison between the text-dependent approaches and the usually more difficult text-independent approaches. Text-independent approaches including Gish's segmental Gaussian model, Reynolds' Gaussian Mixture Model [5], need to deal with unique problems (e.g. sounds and articulations present in the test material but not in training).

It's difficult also to compare the binary choice verification task and the usually more difficult multiple-choice identification task. General trends depict accuracy improvements over time with larger tests i.e. enabled by larger data bases, thus enhancing confidence in performance measurements. These speaker recognition systems need to be used in combination with other authenticators (for e.g. smart cards) in case of high-security applications. The performance of current speaker-recognition systems, however, makes them ideal for a number of practical applications. There exist several commercial ASV systems, including those from Lernout & Hauspie, T-NETIX, Veritel, Voice Control Systems and many others. Perhaps the largest scale deployment of any biometric system to date is Sprint's Voice FONCARD. Speaker-verification applications include access control, telephone credit cards and a lot others. Automatic speaker-recognition systems could help a great deal in reducing crime over fraudulent transactions substantially. However it is imperative to understand the errors made by these ASV systems keeping note of the fact that these systems have gained widespread use across the world. They experience two kinds of errors:- Type I error (False Acceptance of an invalid user (FA)) and Type II error (False Rejection of a valid user(FR)). It uses a pair of subjects: an impostor and a target, to make a false acceptance error. These errors are the ultimate cause of concern in high-security speaker-verification applications.

1.5 THESIS CONTRIBUTION

We have used several types of speaker recognition systems for feature extraction and matching purposes. For this we have initially taken a database of 8 different speakers and recorded 6 samples of the same text speech from each speaker. Then we have extracted Cepstral Coefficients (using DCT) and MFCC or Mel-frequency Cepstral coefficients [2] [3] from their speeches as a part of the feature extraction process. For generating templates and for feature matching purpose we have made use of VQLBG (Vector Quantization using Linde, Buzo and Gray [8]), DTW (Dynamic Time Warping) and GMM (Gaussian Mixture Modeling) algorithms. We then calculated the efficiencies of each algorithm and proposed the best method for robust speaker recognition. Thus we have carried out the task of speaker identification which forms an integral component of speaker recognition. All this work has been carried out using MATLAB 2008.

1.6 OUTLINE OF THESIS

The purpose of the introduction chapter is to provide a general framework for speaker recognition, an overview of the entire thesis, and a note of the previous contributions in the field of speaker recognition.

Chapter 2 contains an introduction to speaker recognition and the various classifications of an automatic speaker recognition system

Chapter 3 contains the pre-processing tasks such as truncation, frame blocking, windowing and calculating FFT coefficients and various speech feature extraction methods we have adopted such as cepstral coefficients using DCT and the cepstrum from the Melfrequency wrapped spectrum which are MFCCs of the speaker.

Chapter 4 contains a description of the various feature matching algorithms used such as Vector Quantization using Linde Buzo and Gray algorithm (VQLBG), Dynamic Time Warping (DTW) and Gaussian Mixture Modeling (GMM).

Chapter 5 contains the results we obtained for feature extraction and feature matching and the efficiencies of each of the algorithms discussed in the previous chapter.

Chapter 6 concludes our project work.

CHAPTER 2

PRINCIPLES OF SPEAKER RECOGNITION

2.1 INTRODUCTION

Speaker recognition is a biometric system which performs the computing task of validating a user's claimed identity using the characteristic features extracted from their speech samples. Speaker identification [6] is one of the two integral parts of a speaker recognition system with speaker verification being the other one. The main difference between the two categories has been explained in this chapter. On a brief note, speaker verification performs a binary decision which consists of determining whether the person speaking is the same person he/she claims to be or to put it in other words verifying their identity. Speaker identification on the other hand does the job of matching (comparing) the voice of the speaker (known or unknown) with a database of reference templates in an attempt to identify the speaker. In our project, speaker identification will be the focus of the research.

Speaker identification is further divided into two subcategories, text dependent and text-independent speaker identification [10]. Text-dependent speaker identification differs quite from text-independent as in the aforementioned identification is done on the voice instance of a specific word, whereas in the latter the speaker can say anything. Our project will consider only the text-dependent speaker identification category.

The field of speaker recognition has gained immense popularity in various applications ranging from embedding recognition in a product which allows a unique level of hands-free and intuitive user interaction, automated dictation and command interfaces etc. The various phases of our project will lead to an in-depth understanding of various speaker recognition models being employed while becoming involved with the speaker recognition community.

2.2 CLASSIFICATION

Speaker recognition can be classified into a number of categories. Figure 2.1 below provides the various classifications of speaker recognition.

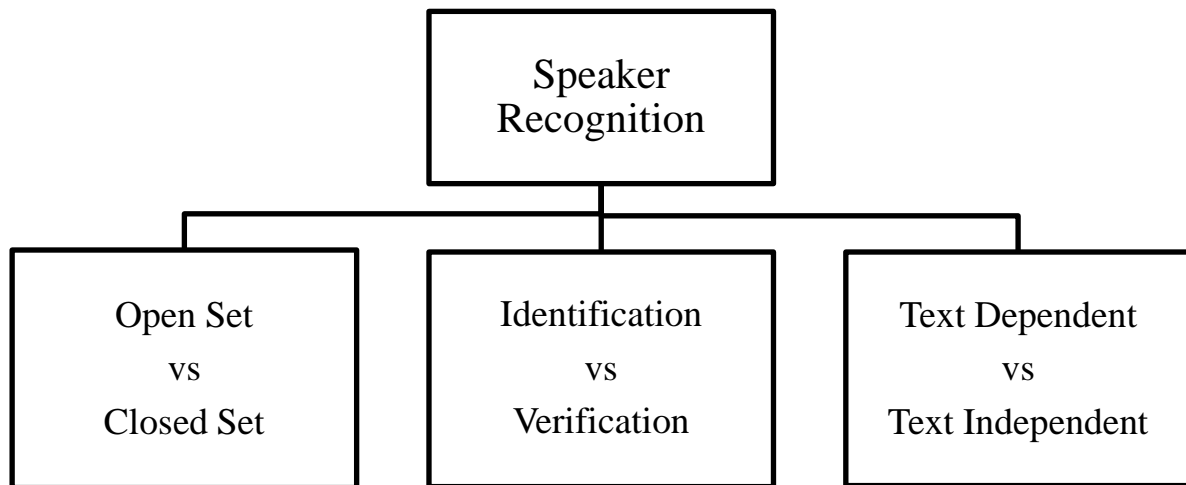


Figure 2.1: Classification of Speaker Recognition

2.2.1 OPEN SET vs CLOSED SET

Speaker recognition can be classified into open set and closed set speaker recognition. This category of classification is based on the set of trained speakers available in a system. Let us discuss them in details.

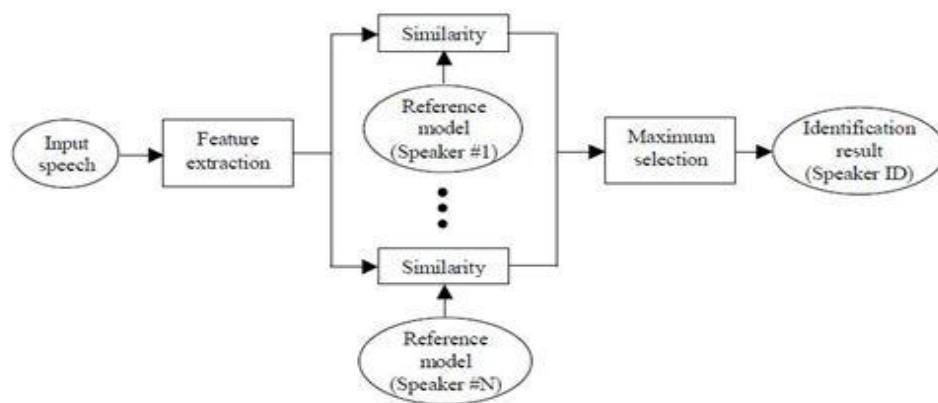
1. **Open Set:** An open set system can have any number of trained speakers. We have an open set of speakers and the number of speakers can be anything greater than one.
2. **Closed Set:** A closed set system has only a specified (fixed) number of users registered to the system.

In our project, we have used a closed set of trained speakers.

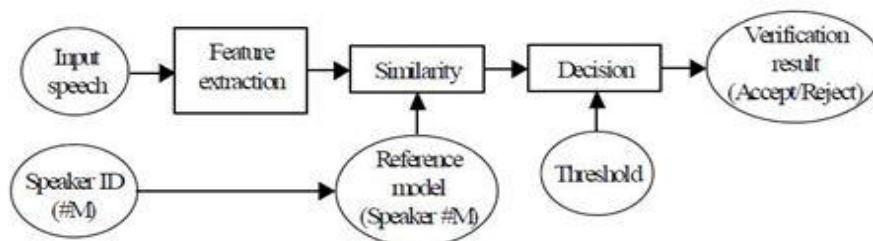
2.2.2 IDENTIFICATION vs VERIFICATION

This category of classification is the most important among the lot. Automatic speaker identification and verification are often considered to be the most natural and economical methods for avoiding unauthorized access to physical locations or computer systems. Let us discuss them in detail:-

1. **Speaker identification:** It is the process of determining which registered speaker provides a given utterance.
2. **Speaker verification:** It is the process of accepting or rejecting the identity claim of a speaker. Figure 2.2 below and figure 2.3 in the next page illustrate the basic differences between speaker identification and verification systems.

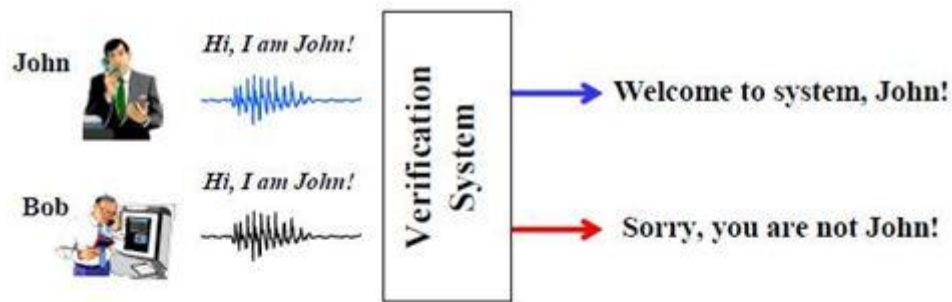


(a) Speaker identification

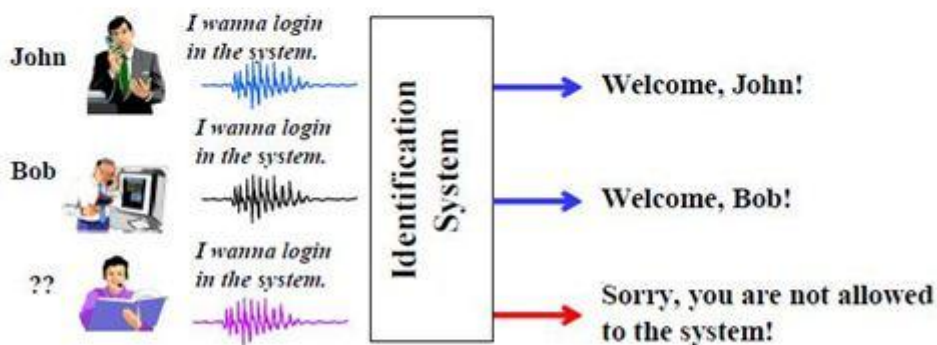


(b) Speaker verification

Figure 2.2: Block Diagrams of Identification and Verification systems



SPEAKER VERIFICATION



SPEAKER IDENTIFICATION

Figure 2.3: Practical examples of Identification and Verification Systems

Both the figures depict the differences between ASI (Automatic Speaker Identification) and ASV (Automatic Speaker Verification) systems. Figure 2.2 gives the theoretical block diagrams of both the processes whereas figure 2.3 gives a practical implementation of the systems. In our project we have focussed only on ASI systems.

2.2.3 TEXT-DEPENDENT vs TEXT-INDEPENDENT

This is another category of classification of speaker recognition systems. This category is based upon the text uttered by the speaker during the identification process. Let us discuss each in details:-

1. **Text-Dependent:** In this case, the test utterance is identical to the text used in the training phase. The test speaker has prior knowledge of the system.
2. **Text-Independent:** In this case, the test speaker doesn't have any prior knowledge about the contents of the training phase and can speak anything.

In our project, we have used the text-dependent model. Thus, we have designed a closed-set text-dependent ASI (Automatic Speaker Identification) system in our thesis work.

2.3 MODULES

This section describes the two main modules of a speaker recognition system (in our case an ASI system).Figure 2.4 below provides the various modules and some of the methods we have used in our project work.

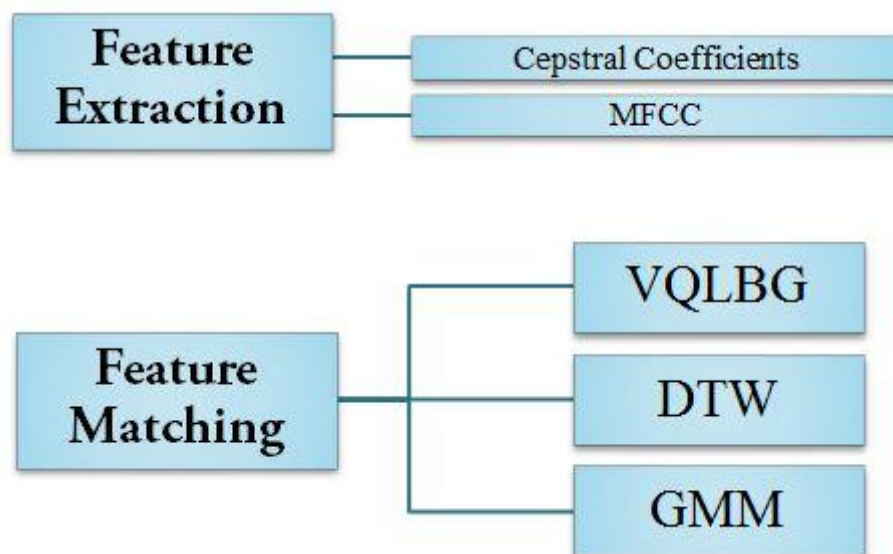


Figure 2.4: Modules of a speaker recognition system

Let's discuss the 2 modules.

1. **Feature Extraction:** The purpose of this module is to convert the speech waveform into a set of features or rather feature vectors used for further analysis.
2. **Feature Matching:** In this module the features extracted from the input speech are matched with the stored template (reference model) and a recognition decision is made.

As shown in the previous figure, we have made use of Cepstral Coefficients using DCT and MFCC for feature extraction process. For feature matching, we have used 3 algorithms namely VQLBG, DTW and GMM.

2.4 INITIALS

This section describes the initial phase of the project which includes the following:-

1. Generating Database

We generated a database of voice signals by allowing 8 persons to utter the word "Hello" and recording their voices in .wav format.

2. Reading an Audio Clip

The audio samples were read in MATLAB using the '**wavread**' command. It was found out that the sampling frequency for the command is fixed at 44100 Hz.

3. Audio Playback

The audio files were played by using '**play**' command upon an 'audioplayer' object generated from the original signal. Also by varying the sampling frequency used, we can have faster and slower versions of the same audio clip.

2.5 SUMMARY

In this chapter, we have discussed the various classifications and modules of an automatic speaker recognition system. The next chapter will focus on the feature extraction techniques.

CHAPTER 3

FEATURE EXTRACTION

3.1 INTRODUCTION

The purpose of this module is to convert the speech waveform into a set of features or rather feature vectors (at a considerably lower information rate) for further analysis. This is often referred to as the signal-processing front end.

The speech signal is called a quasi-stationary signal i.e. a slowly time-varying signal. An example of a speech signal is shown in Figure 3.1.

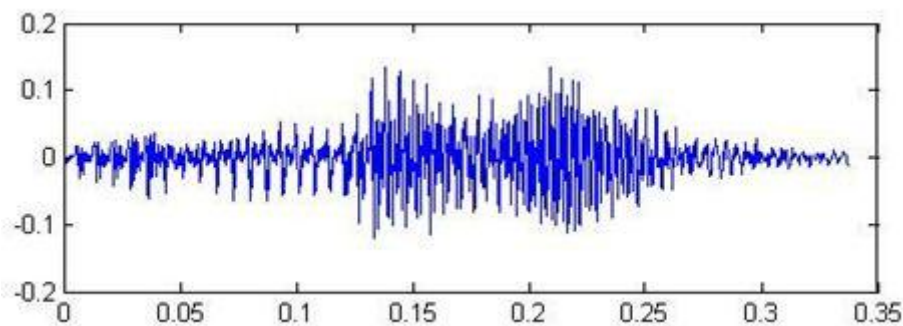


Figure 3.1: Example of a Speech Signal

When examined over a short period of time (for e.g. between 5 and 100 ms), the characteristics of the speech signal are found to be fairly stationary. However, the signal characteristics tend to change over longer periods of time (on the order of 1/5 seconds or even more). This reflects the different speech utterances being spoken. Therefore, short-time spectral analysis is the most common way to characterize a speech signal [11].

A wide range of possible methods exist for parametrically representing the speech signal for the speaker recognition purpose. These include Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), Cepstral Coefficients using DCT and several others. In our project we have used Cepstral Coefficients using DCT and MFCC processor.

3.2 PRE-PROCESSING

Before extracting the features of the signal various pre-processing tasks must be performed. The speech signal needs to undergo various signal conditioning steps before being subjected to the feature extraction methods. These tasks include:-

- Truncation
- Frame blocking
- Windowing
- Short term Fourier Transform

3.2.1 TRUNCATION

The default sampling frequency of wavread command is 44100 Hz. When an audio clip is recorded, say for duration of 2 secs, the number of samples generated would be around 90000 which are too much to handle. Hence we can truncate the signal by selecting a particular threshold value. We can mark the start of the signal where the signal goes above the value while traversing the time axis in positive direction. In the same we can have the end of the signal by repeating the above algorithm in the negative direction. Figure 3.2 below shows the result obtained on truncating a voice signal.

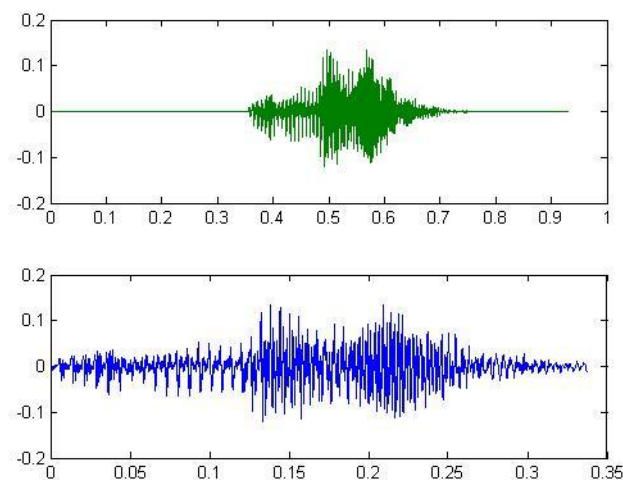


Figure 3.2: Truncated version of original signal

3.2.2 FRAME BLOCKING

In this step the continuous speech signal is divided into frames of N samples, with adjacent frames being separated by M samples with the value M less than that of N . The first frame consists of the first N samples. The second frame begins from M samples after the first frame, and overlaps it by $N - M$ samples and so on. This process continues until all the speech is accounted for using one or more frames [11]. We have chosen the values of M and N to be $N = 256$ and $M = 128$ respectively. Figure 3.3. below gives the frame output of the truncated signal.

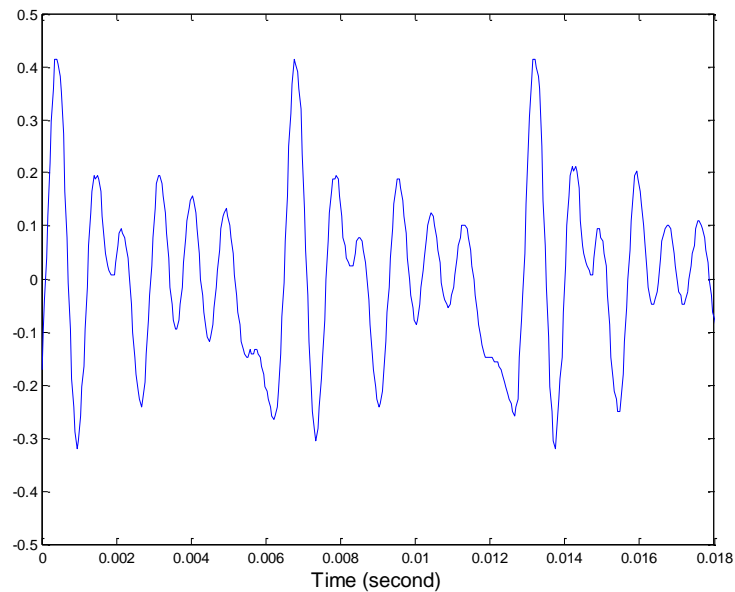


Figure 3.3: Frame output of truncated signal

The value of N is chosen to be 256 because the speech signal is assumed to be periodic over the period. Also the frame of length 256 being a power of 2 can be used for using a fast implementation of Discrete Fourier Transform (DFT) called the FFT (Fast Fourier Transform).

3.2.3 WINDOWING

The next step is to window each individual frame to minimize the signal discontinuities at the beginning and end of each frame. The concept applied here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n)$, $0 \leq n \leq N-1$, where N is the frame length, then the result of windowing is the signal

$$y(n) = x(n)w(n), \quad 0 \leq n \leq N-1$$

We have used the Hamming window in our project, which has the form:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

Figure 3.4 below gives the figure of a Hamming window.

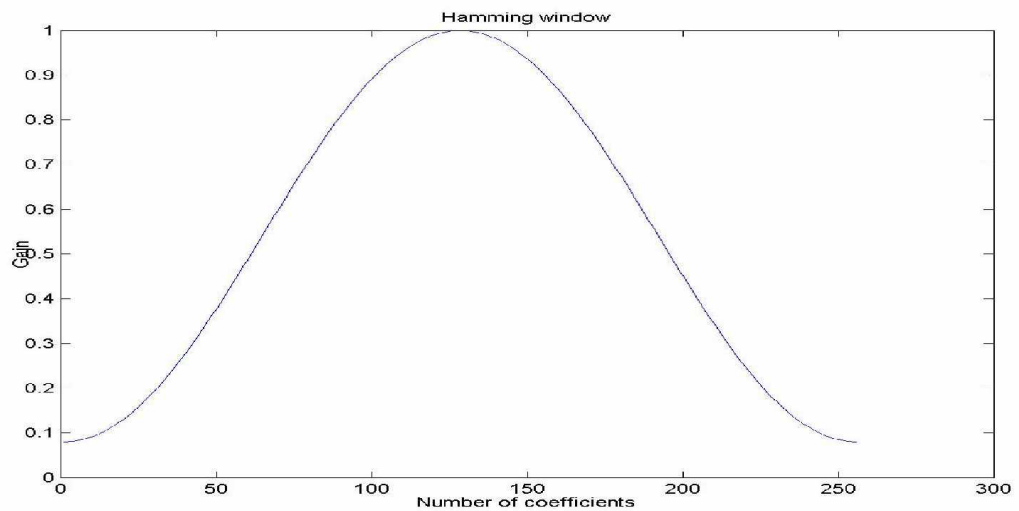


Figure 3.4: Hamming Window

3.2.4 SHORT TERM FAST FOURIER TRANSFORM

The next step is the application of Fast Fourier Transform (FFT), which converts each frame of N samples from the time domain into the frequency domain. The FFT which is a fast algorithm to implement the Discrete Fourier Transform (DFT) is defined on the set of N samples $\{x_n\}$, as follows:-

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \quad k = 0, 1, 2, \dots, N-1$$

In general X_k 's are complex numbers and we consider only their absolute values. The resulting sequence $\{X_k\}$ is interpreted as follows: positive frequencies $0 \leq f < F_s/2$ correspond to values $0 \leq n \leq N/2 - 1$, while negative frequencies $-F_s/2 < f < 0$ correspond to $N/2 + 1 \leq n \leq N - 1$. F_s denotes the sampling frequency. The result after this step is often referred to as spectrum or periodogram [11].

3.3 CEPSTRAL COEFFICIENTS USING DCT

Pre-processing the signal reduces the computational complexity while operating on the speech signal. We reduce the number of samples of operation. Instead of working on a large set of samples we restrict our operation to a frame of sufficiently reduced length. After conditioning the speech signal i.e. after pre-processing the next step is to extract the features of the training signal. We have made use of 2 methods for the same. The first method is calculating the Cepstral Coefficients of the signal using DCT (Discrete Cosine Transform). The Cepstral Coefficients are calculated using the following formula:-

$$\text{ceps} = \text{dct}(\log(\text{abs}(\text{FFT}(y_{\text{windowed}}))))$$

The 1st 10 cepstral coefficients are taken and averaged over 3 training signals belonging to a particular user and represented in a table using MATLAB. The results of the same will be shown in Chapter 5.

3.4 MFCC (MEL FREQUENCY CEPSTRAL COEFFICIENTS)

MFCC (Mel Frequency Cepstral Coefficients) Calculation in the second feature extraction method we have used. Figure 3.5 below gives the block diagram of a MFCC processor. MFCC's are based on the known variation of the human ear's critical bandwidths with frequency. Filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in mel-frequency scale, which is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz [11].

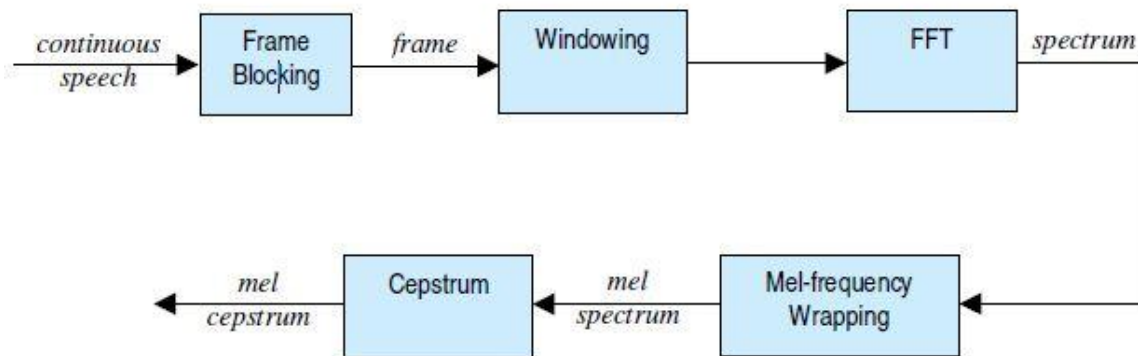


Figure 3.5: MFCC Processor

3.4.1 MEL-FREQUENCY WRAPPING

As given in the block diagram we have already subjected the continuous speech signal to frame blocking, windowing and FFT in the pre-processing step. The result of the later step is the spectrum of the signal.

Psychophysical studies have revealed that human perception of frequency content of sounds for speech signals doesn't follow a linear scale. For each tone with an actual frequency f , a subjective pitch is measured on a scale called the 'mel' scale. The mel-frequency scale provides a linear frequency spacing below 1 KHz and a logarithmic spacing above 1 KHz [1] [2].

The Mel Frequency Scale is given by:-

$$F_{\text{mel}} = (1000/\log(2)) * \log(1 + f/1000)$$

One approach towards simulating the subjective spectrum is to use a filter bank which is spaced uniformly on the mel-scale. The filter bank has a triangular band pass frequency response. The spacing and the bandwidth is determined by a constant mel frequency interval. We choose K, the number of mel spectrum coefficients to be 20. This filter bank being applied in the frequency domain simply amounts to applying the triangle-shape windows to the spectrum. A useful way to think about this filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain. Figure 3.5 below gives an example of a mel-spaced frequency bank.

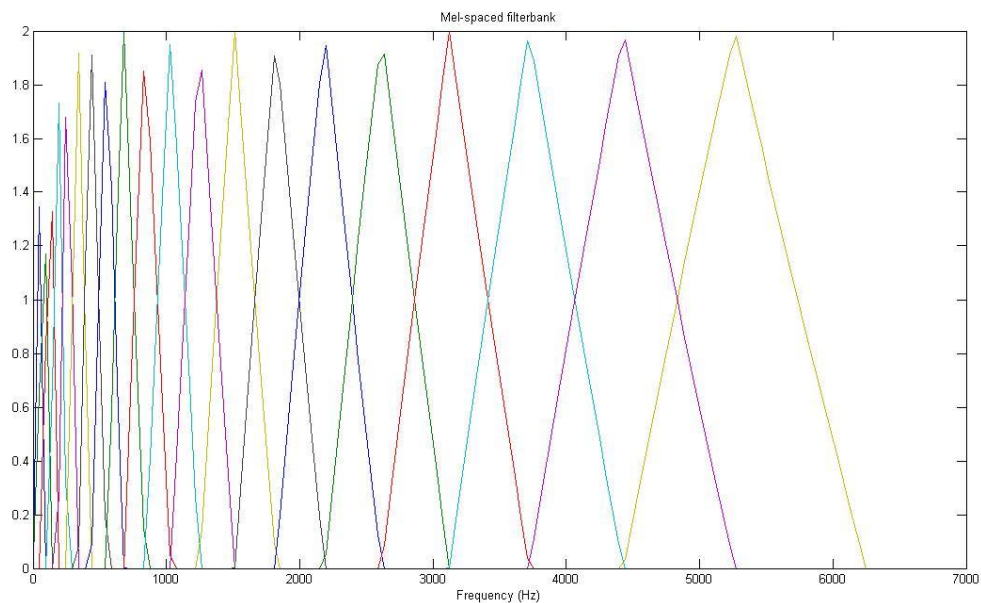


Figure 3.6: Example of a Mel-spaced frequency bank

We have used 20 filters in our case as shown in the figure above.

3.4.2 CEPSTRUM

In this final step, we convert the log Mel spectrum to time domain. The result is called the MFCC (Mel Frequency Cepstral Coefficients). This representation of the speech spectrum provides a good approximation of the spectral properties of the signal for the given frame analysis. The Mel spectrum coefficients being real numbers are then converted to time domain using Discrete Cosine Transform (DCT). If we denote the Mel power spectrum coefficients that are the result of the last step as S_k , $k = 1, 2, \dots, K$, we can calculate the MFCC's C_n as

$$c_n = \sum_{k=1}^K (\log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad n = 1, 2, \dots, K$$

We exclude the first component from the DCT since it represents the mean value of the input signal which carries little speaker specific information [11].

3.5 SUMMARY

By applying the procedure described above, a set of mel-frequency cepstrum coefficients (MFCC) is computed for each speech frame of around 30 secs. The set of coefficients is called an acoustic vector. Thus each input speech utterance is transformed into a sequence of acoustic vectors.

CHAPTER 4

FEATURE MATCHING

4.1 INTRODUCTION

The problem of speaker recognition has always been a much wider topic in engineering field so called pattern recognition. The aim of pattern recognition lies in classifying objects of interest into a number of categories or classes. The objects of interest are called patterns and in our case are sequences of feature vectors that are extracted from an input speech using the techniques described in the previous chapter. Each class here refers to each individual speaker. Since here we are only dealing with classification procedure based upon extracted features, it can also be abbreviated as feature matching.

To add more, if there exists a set of patterns for which the corresponding classes are already known, then the problem is reduced to supervised pattern recognition. These patterns are used as training set and classification algorithm is determined for each class. The rest patterns are then used to test whether the classification algorithm works properly or not; collection of these patterns are referred as the test set. In the test set if there exists a pattern for which no classification could be derived, and then the pattern is referred as unregistered user for the speaker identification process. In real time environment the robustness of the algorithm can be determined by checking how many registered users are identified correctly and how efficiently it discards the unknown users.

Feature matching problem has been sorted out with many class-of-art efficient algorithms like VQLBG, DTW and stochastic models such as GMM, HMM. In our study project we have put our focus on VQLBG, DTW and GMM algorithm. VQLBG algorithm due to its simplicity has been stressed at the beginning followed by DTW and GMM respectively.

4.2 SPEAKER MODELING

Using Cepstral coefficients and MFCC as illustrated in the previous section, a spoken syllable can be represented as a set of feature vectors. A person uttering the same word but at a different time instant will be having similar still differently arranged feature vector sequence. The purpose of voice modeling lies in building a model that can capture these variations in a set of features extracted from a given speaker. There are usually two types of models those are extensively used in speaker recognition systems:

- **Stochastic models**
- **Template models**

The stochastic model exploits the advantage of probability theory by treating the speech production process as a parametric random process. It assumes that the parameters of the underlying stochastic process can be estimated precisely, in a well-defined manner. In parametric methods usually assumption is made about generation of feature vectors but the non-parametric methods are free from any assumption about data generation. The template model (non-parametric method) attempts to generate a model for speech production process for a particular user in a non-parametric manner. It does so by using sequences of feature vectors extracted from multiple utterances of the same word by the same person. Template models used to dominate early work in speaker recognition because it works without making any assumption about how the feature vectors are being formed. Hence the template model is intuitively more reasonable. However, recent work in stochastic models has revealed them to be more flexible, thus allowing for generation of better models for speaker recognition process. The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Gaussian Mixture Modeling (GMM), and Vector Quantization (VQ). In a speaker recognition system, the process of representing each speaker in an unique and efficient manner is known as vector quantization. It is the process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is abbreviated a cluster and represented by its center called a code word. A codebook is collection of all code words. Hence for multiple users there should be multiple codebooks each representing the corresponding speaker. The data is thus significantly compressed, yet still accurately represented. Without quantization of the feature vectors, computational complexity of a system would be very large as there would be large number of feature vectors

present in multi-dimensional space. In a speaker recognition system, feature vectors are usually contained in vector space, which are obtained from the feature extraction described above. When vector quantization process goes to completion, only remnants are a few representative vectors, and these vectors are collectively known as the speaker's codebook. The codebook then serves as template for the speaker, and is used when testing a speaker in the system [11].

4.3 VECTOR QUANTIZATION

Vector quantization (VQ in short) involves the process of taking a large set of feature vectors of a particular user and producing a smaller set of feature vectors that represent the centroids of the distribution, i.e. points spaced so as to minimize the average distance to every other point. Vector quantization is used since it would be highly impractical to represent every single feature vector in feature space that we generate from the training utterance of the corresponding speaker. While the VQ algorithm does take a while to generate the centroids, it saves a lot of time during the testing phase as we are only considering few feature vectors instead of overloaded feature space of a particular user. Therefore is an economical compromise that we can live with. A vector quantizer maps k -dimensional vectors in the vector space R^k into a finite set of vectors $Y = \{y_i; i = 1, 2, \dots, N\}$. Here k -dimension refers to the no of feature coefficients in each feature vector. Each vector y_i is called a code vector or a codeword and the set of all the codewords is called a codebook. Hence for a given number of users, code books are generated for each speaker during the training phase using VQ method. For each codeword y_i , there is a 'nearest neighbor' region associated with it called Voronoi region [11], and is defined by

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \text{ for all } j \neq i\}$$

The set of voronoi regions partition the entire feature space of a given user such that-

$$\bigcup_{i=1}^N V_i = R^k$$

$$\bigcap_{i=1}^N V_i = \phi$$

for all $i \neq j$

To illustrate what a Voronoi region really is, we consider feature vectors having only two feature coefficients. Thus they can be each represented in two dimensional feature space. Figure 4.1 in the next page shows all the feature vectors extracted from a given user in a two dimensional vector space. As shown in the figure, all feature vectors have been associated to their nearest neighbor using vector quantization process and corresponding centroids have been generated. The centroids are shown using red dot and the numerous green dots represent the feature vectors. Each codeword (centroid) resides in its own Voronoi region. These regions are separated by imaginary lines in the figure for visualization. Given an input vector, Euclidian distance is calculated from each codeword and the one having least Euclidian distance is the appropriate codeword for that given vector. The Voronoi region associated with the given centroid is cluster region for the vector. The Euclidean distance is defined by:-

$$d(x, y_i) = \sqrt{\sum_{j=1}^k (x_j - y_{ij})^2}$$

where x_j is the j th component of the input vector, and y_{ij} is the j th component of the codeword y_i .

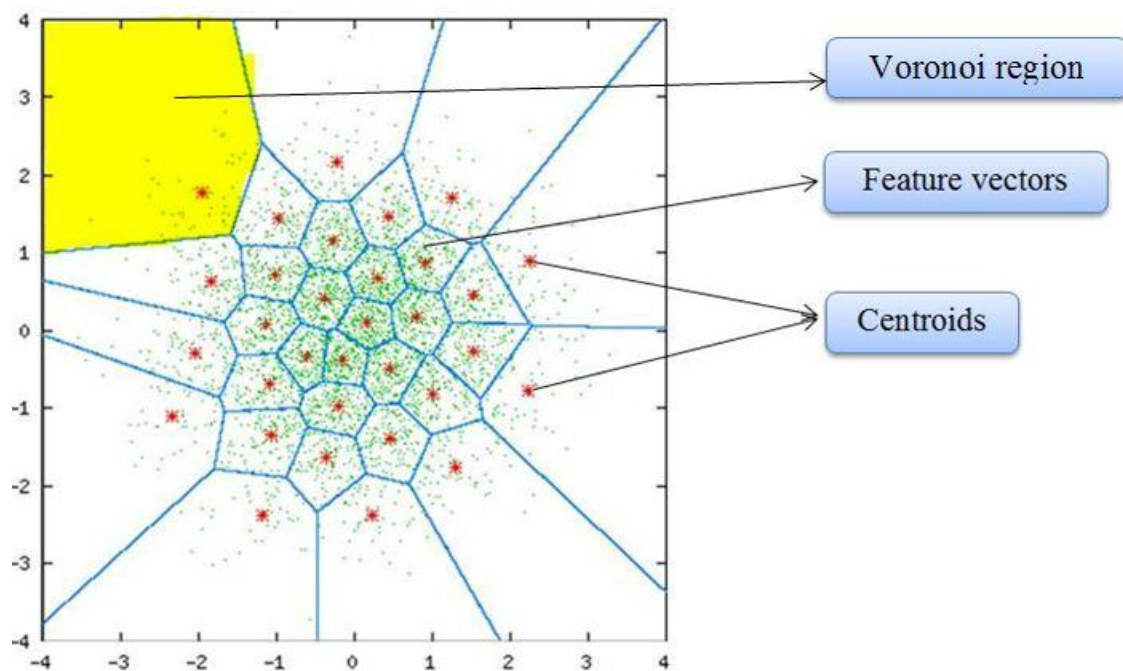


Figure 4.1: Codewords in 2-dimensional space. Input vectors marked with green dots, codewords marked with red stars, and the Voronoi regions separated by boundary lines

The key advantages of VQ are

- Reduced storage for spectral analysis information
- Reduced computation for determining similarity of spectral analysis vectors. In speech recognition, a major component of the computation is the determination of spectral similarity between a pair of vectors. Based on the VQ representation this is often reduced to a table lookup of similarities between pairs of codebook vectors.
- Discrete representation of speech sounds

Figure 4.2 below shows the block diagram of a speaker recognition model using VQ:-

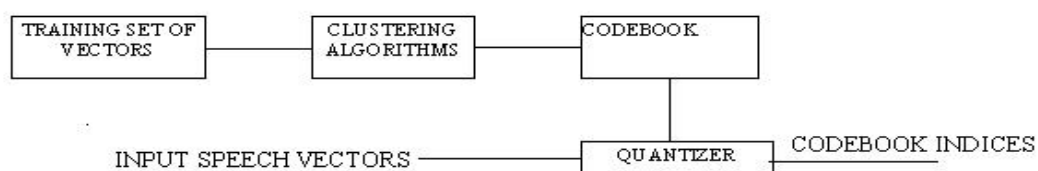


Figure 4.2: Block Diagram of the basic VQ Training and classification structure

Speaker recognition can be done using the code book generated for each registered user via VQ method. Figure 4.3 below best describes the process:-

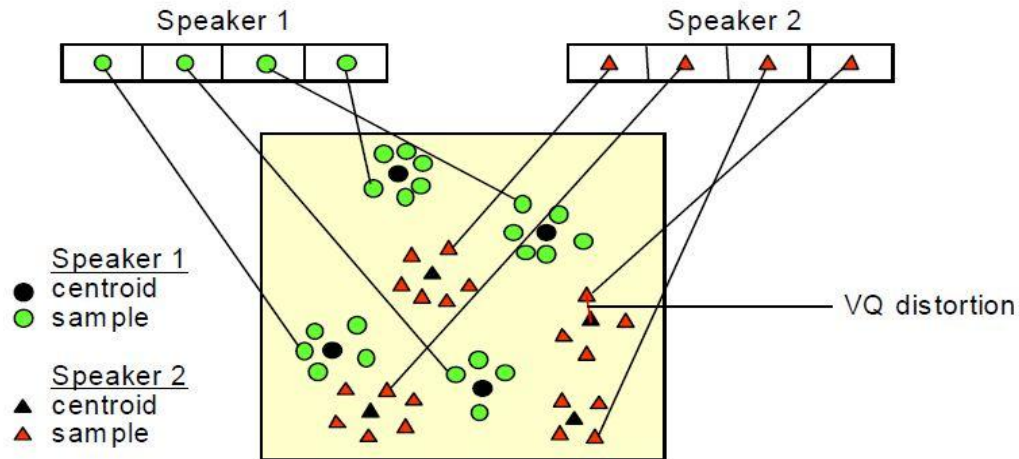


Figure 4.3. Conceptual diagram illustrating vector quantization codebook formation.

As shown in the figure, there are only two registered users for which feature vectors have been extracted and VQ has been performed. For each user only four codewords are generated. These codewords collectively represent the codebook for each user. For both users corresponding code books are thus generated. During the testing phase, feature vectors of unknown users are mapped to feature space assuming the unknown user to be one of the trained speakers. Then for each feature vector the nearest code word is found using Euclidian distance for a given code book. The least distance found is termed as VQ distortion. In a similar manner VQ distortions are calculated for the rest feature vectors and summed up. This procedure is repeated for the second speaker. The least sum of the VQ distortions for each user will give the desired user.

4.4 OPTIMISATION USING LBG ALGORITHM

After the enrolment session, the feature vectors extracted from input speech of each speaker provide a set of training vectors for that speaker. As explained in the previous section, the next important task is to build a speaker-specific VQ codebook for each speaker using the training vectors extracted. There is a well-known algorithm, namely LBG algorithm [Linde, Buzo and Gray, 1980], for clustering a set of L training vectors into a set of M codebook vectors. The algorithm is formally implemented by the following recursive procedure:-

1. Design a 1-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Increase the size of the codebook twice by splitting each current codebook \mathbf{y}_n according to the rule

$$\mathbf{y}_n^+ = \mathbf{y}_n(1 + \varepsilon)$$

$$\mathbf{y}_n^- = \mathbf{y}_n(1 - \varepsilon)$$

where n varies from 1 to the current size of the codebook, and ε is a splitting parameter (we choose $\varepsilon=0.01$).

3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is the closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).
4. Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold
6. Iteration 2: repeat steps 2, 3 and 4 until a codebook size of M is designed.

Intuitively, the LBG algorithm generates an M -vector codebook iteratively. It starts first by producing a 1-vector codebook, then uses a splitting technique on the codeword to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M -vector codebook is obtained [11].

Figure 4.4 below shows, in a flow chart, the detailed steps of the LBG algorithm. “Cluster vectors” is the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. “Find centroids” is the procedure for updating the centroid. In the nearest-neighbor search “Compute D (distortion)” sums the distances of all training vectors so as to decide whether the procedure has converged or not. Decision boxes are there to terminate the process.

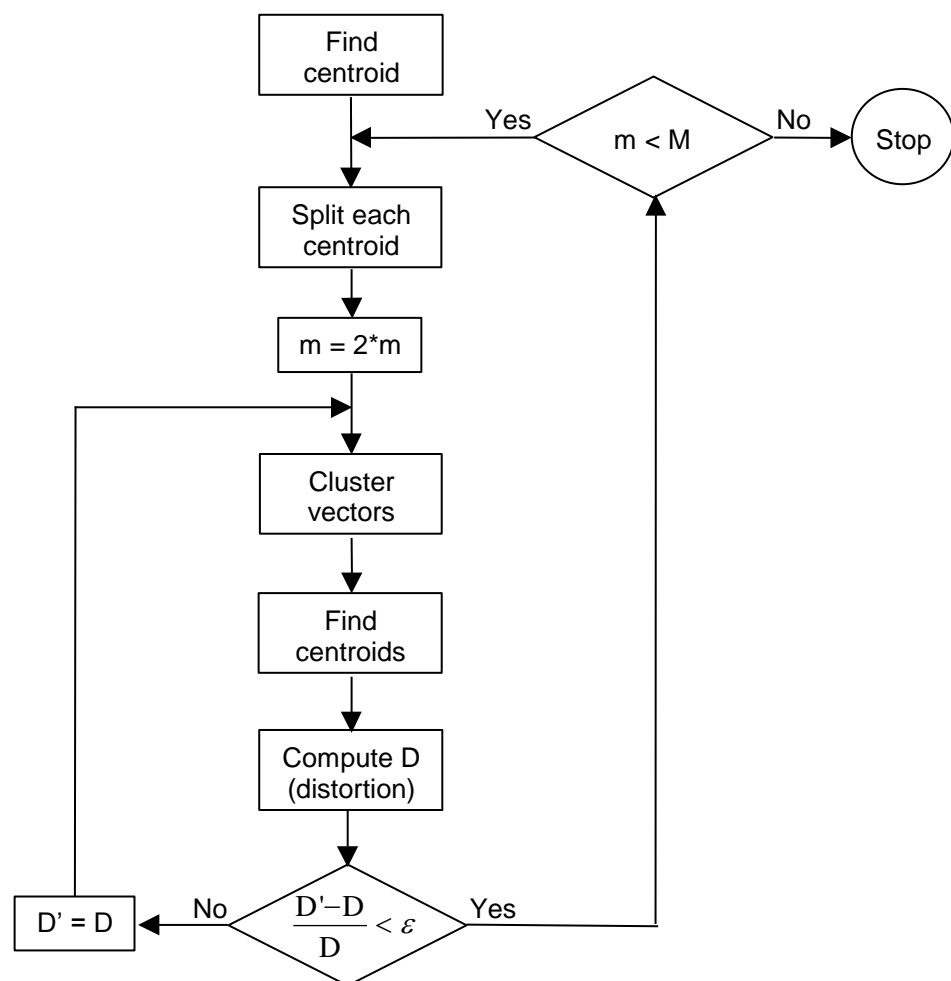


Figure 4.4: Flow diagram of the LBG algorithm

4.5 DTW (DYNAMIC TIME WARPING) ALGORITHM

4.5.1 INTRODUCTION

Time series is the most frequent form of data occurring in every field of science. A common task with time series data is comparing one sequence with another. Many a time simply Euclidian distance suffices the task. Problem arises when two sequences have the approximately the same overall component shapes, but these shapes do not line up in X-axis. Any distance (Euclidean, Manhattan, ...) which aligns the i -th point on one time series with the i -th point on the other will produce a poor similarity score. A non-linear (elastic) alignment produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis. DTW is such an algorithm which exploits non-linear modeling to get efficient warping of one time series with respect to another. Figure 4.5 below illustrates the warping procedure:-

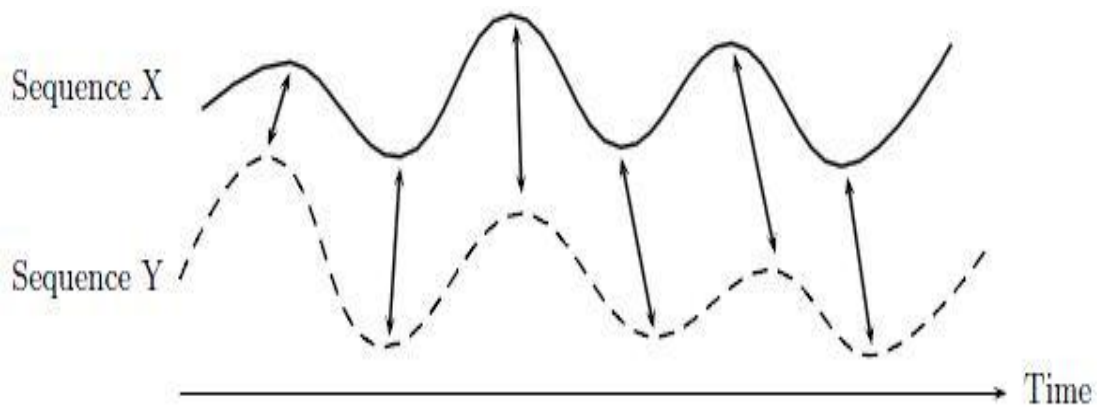


Figure 4.5: Dynamic Time Warping

Dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. Although it is not as efficient as HMM or GMM processes still its easy hardware compatibility has made it most preferable choice for mobile applications. To add more the training procedure in DTW algorithm is faster as compared to other parametric or non-parametric methods.

In our project work we have only considered classical DTW. As our data base contains speaker data which are very large in size, thus instead of going for each data point we used the feature vectors of each speaker to find optimal matching among them. These feature vectors have already been derived using MFCC feature extraction process.

4.5.2 CLASSICAL DTW ALGORITHM

Suppose we have two time series Q and C , of length n and m respectively, where:-

$$Q = q_1, q_2, \dots, q_i, \dots, q_n$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m$$

To align the two sequences using DTW we construct a n -by- m matrix where the $(i^{\text{th}}, j^{\text{th}})$ element of the matrix contains the distance $d(q_i, c_j)$ between the two points q_i and c_j (Typically the Euclidean distance is used, so $d(q_i, c_j) = (q_i - c_j)^2$). Each matrix element (i, j) corresponds to the alignment between the points q_i and c_j . A warping path W is a continuous (in the sense stated below) set of matrix elements that defines a mapping between Q and C . The k^{th} element of W is defined as $w_k = (i, j)_k$ so we have:-

$$W = w_1, w_2, \dots, w_k, \dots, w_K \quad \max(m, n) \leq K < m+n-1$$

The warping path is typically subject to several constraints:-

1. **Boundary conditions:** $w_1 = (1, 1)$ and $w_K = (m, n)$, simply stated, this requires the warping path to start and finish in diagonally opposite corner cells of the matrix.
2. **Continuity:** Given $w_k = (a, b)$ then $w_{k-1} = (a', b')$ where $a-a' \leq 1$ and $b-b' \leq 1$. This restricts the allowable steps in the warping path to adjacent cells (including diagonally adjacent cells).
3. **Monotonicity:** Given $w_k = (a, b)$ then $w_{k-1} = (a', b')$ where $a-a' \geq 0$ and $b-b' \geq 0$. This forces the points in W to be monotonically spaced in time.

The numbers of warping paths that satisfy the above conditions are exponentially large; however we are interested only in the particular path which minimizes the warping cost:-

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} / K \right\}$$

The K in the denominator is used to compensate for the fact that warping paths may have different lengths [12]. This path can be found very efficiently using dynamic programming.

4.5.3 DYNAMIC PROGRAMMING

Optimal path providing the best wrap between a test utterance and a given speaker can be efficiently determined using dynamic programming. In this method we will generate a cumulative distance matrix out of Euclidian distance matrix already obtained. The process of generating the cumulative matrix can be stated as follows:-

1. Use the constellation given below.

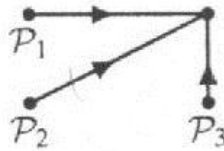


Figure 4.6: Constellation diagram

Here P₁, P₂, P₃ refer to three different paths. The path which has the least Euclidian distance is considered to be the best path.

2. Take the initial condition $g(1, 1) = d(1, 1)$ where g and d are cumulative distance matrix and Euclidian matrix respectively.
3. Calculate the first row $g(i, 1) = g(i-1, 1) + d(i, 1)$.
4. Calculate the first column $g(1, j) = g(1, j) + d(1, j)$.
5. Move to the second row $g(i, 2) = \min(g(i, 1), g(i-1, 1), g(i-1, 2)) + d(i, 2)$. Note for each cell the index of this neighboring cell, which contributes the minimum score (red arrows).

6. Carry on from left to right and from bottom to top with the rest of the grid

$$g(i, j) = \min(g(i, j-1), g(i-1, j-1), g(i-1, j)) + d(i, j).$$
7. Find out the value of $g(n, m)$.

Figure 4.7 below illustrates the above algorithm:-

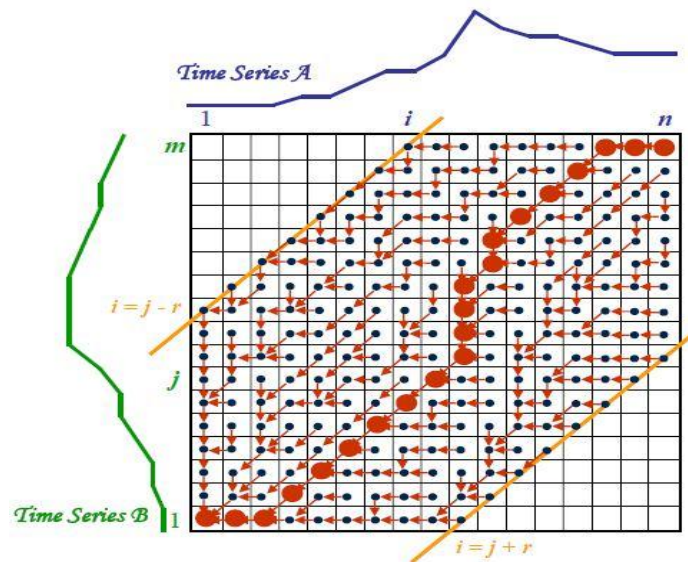


Figure 4.7: Cumulative matrix of two time series data

4.5.4 SPEAKER IDENTIFICATION

Let's consider that we have a set of speakers whose feature vectors are already known. When we have an identification task of an unknown speaker assuming that the speaker is one of speakers whose voice samples we already have, we first go for feature extraction. When we have acquired the feature vectors then we try to warp the unknown vectors with respect to a reference speaker (feature vectors). We follow dynamic programming and calculate the value of $g(n, m)$. We repeat this procedure for all the available speakers. Least value of $g(n, m)$ gives the identity of the unknown speaker.

4.6 GMM (GAUSSIAN MIXTURE MODELLING)

4.6.1 INTRODUCTION

This is one of the non-parametric methods for speaker identification. When feature vectors are displayed in d-dimensional feature space after clustering, they some-how resemble Gaussian distribution. It means each corresponding cluster can be viewed as a Gaussian probability distribution and features belonging to the clusters can be best represented by their probability values. The only difficulty lies in efficient classification of feature vectors. The use of Gaussian mixture density for speaker identification is motivated by two facts [13]. They are:-

- 1- Individual Gaussian classes are interpreted to represents set of acoustic classes. These acoustic classes represent vocal tract information.
- 2- Gaussian mixture density provides smooth approximation to distribution of feature vectors in multi-dimensional feature space [13]

Figure 4.8 gives a better understanding of what GMM really is:-

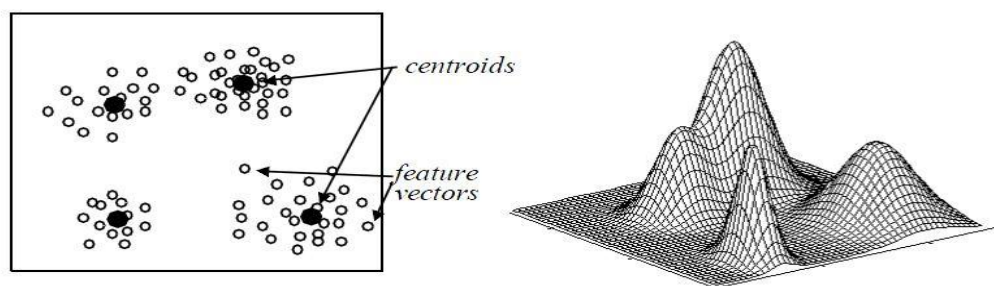


Figure 4.8: GMM model showing a feature space and corresponding Gaussian model

4.6.2 MODEL DESCRIPTION

A Gaussian mixture density is weighted sum of M component densities and given by the equation:-

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^M p_i b_i(\vec{x})$$

where \vec{X} refers to a feature vector, p_i stands for mixture weight of i^{th} component and $b_i(\vec{X})$ is the probability distribution of the i^{th} component in the feature space. As the feature space is D-dimensional, the probability density function $b_i(\vec{X})$ is a D-variate distribution [13]. It is given by the expression:-

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\}$$

where μ_i is the mean of i^{th} component and Σ_i is the co-variance matrix [13].

The complete Gaussian mixture density is represented by mixture weights, mean and co-variance of corresponding component and denoted as:-

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\} \quad i = 1, \dots, M.$$

Diagrammatically it can be shown as:-

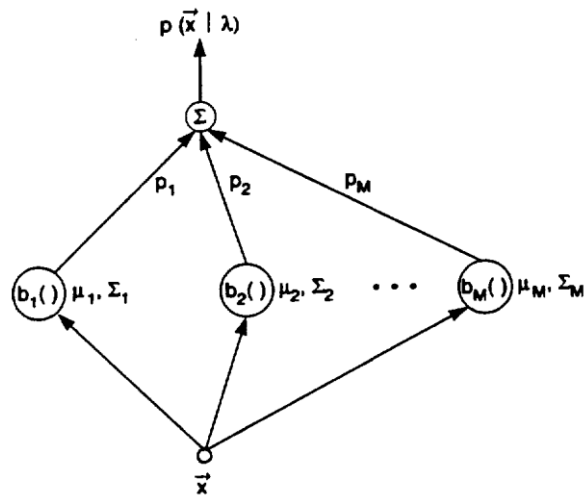


Figure 4.9: Description of M-component Gaussian densities

4.6.3 MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

After obtaining the feature vectors the next task lies in classifying them to different Gaussian components. But initially we don't know mean, co-variance of components present. Thus we can't have proper classification of the vectors. To maximize the classification process for a given set of feature vectors an algorithm is followed known as Expectation Maximization (EM) [14]. This algorithm works as follows:-

1. We assume initial values of μ_i , Σ_i and w_i .
2. Then we calculate next values of mean, co-variance and mixture weights iteratively using the following formula so that probability of classification of set of T feature vectors is maximized.

The following formulae are used:-

Mixture Weights:

$$\bar{p}_i = \frac{1}{T} \sum_{t=1}^T p(i | \vec{x}_t, \lambda)$$

Means:

$$\vec{\mu}_i = \frac{\sum_{t=1}^T p(i | \vec{x}_t, \lambda) \vec{x}_t}{\sum_{t=1}^T p(i | \vec{x}_t, \lambda)}$$

Variances:

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^T p(i | \vec{x}_t, \lambda) x_t^2}{\sum_{t=1}^T p(i | \vec{x}_t, \lambda)} - \bar{\mu}_i^2$$

where $p(i | \vec{X}_t, \lambda)$ is called posteriori probability and is given by the expression:-

$$p(i | \vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^M p_k b_k(\vec{x}_t)}.$$

4.6.4 SPEAKER IDENTIFICATION

After modeling each user's Gaussian mixture density, we have a set of models, each representing Gaussian distribution of all the components present. For K number of speakers it is denoted as $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_k\}$. The objective culminates in finding the speaker model λ having maximum posteriori probability for a given test utterance [13]. Mathematically it can be represented as:-

$$\hat{S} = \arg \max_{1 \leq k \leq S} \Pr(\lambda_k | X) = \arg \max_{1 \leq k \leq S} \frac{p(X | \lambda_k) \Pr(\lambda_k)}{p(X)}$$

4.7 SUMMARY

This chapter gives an account of the various feature matching algorithms we have used. The next chapter contains the results and efficiencies of each method.

CHAPTER 5

RESULTS

5.1 OVERVIEW

Our thesis work is based on identifying an unknown speaker given a set of registered speakers. Here we have assumed the unknown speaker to be one of the known speakers and tried to develop a model to which it can best fit into. In the first step of generating the speaker recognition model, we went for feature extraction using two processes given below:-

1. Cepstral coefficients
2. Mel Frequency Cepstral Coefficients

These features act as a basis for further development of the speaker identification process. Next we went for feature mapping using the following algorithm:-

1. Vector Quantization using LBG (VQLBG)
2. Dynamic Time Warping (DTW)
3. Gaussian Mixture Modeling (GMM)

The results obtained using all the feature extraction and mapping methods are shown in subsequent sections.

5.2 FEATURE EXTRACTION

5.2.1 CEPSTRAL COEFFICIENTS

When we sample a spoken syllable, we'll be having many samples. Then we try to extract features from these sampled values. Cepstral coefficients calculation is one of such methods. Here we initially derive Short Term Fourier Transform of sampled values, then take their absolute value (they can be complex) and calculate log of these absolute values. There after we go for converting back them to time domain using Discrete Cosine Transform (DCT). We have done it for five users and first ten DCT coefficients are Cepstral coefficients. The result obtained is shown in the next page:-

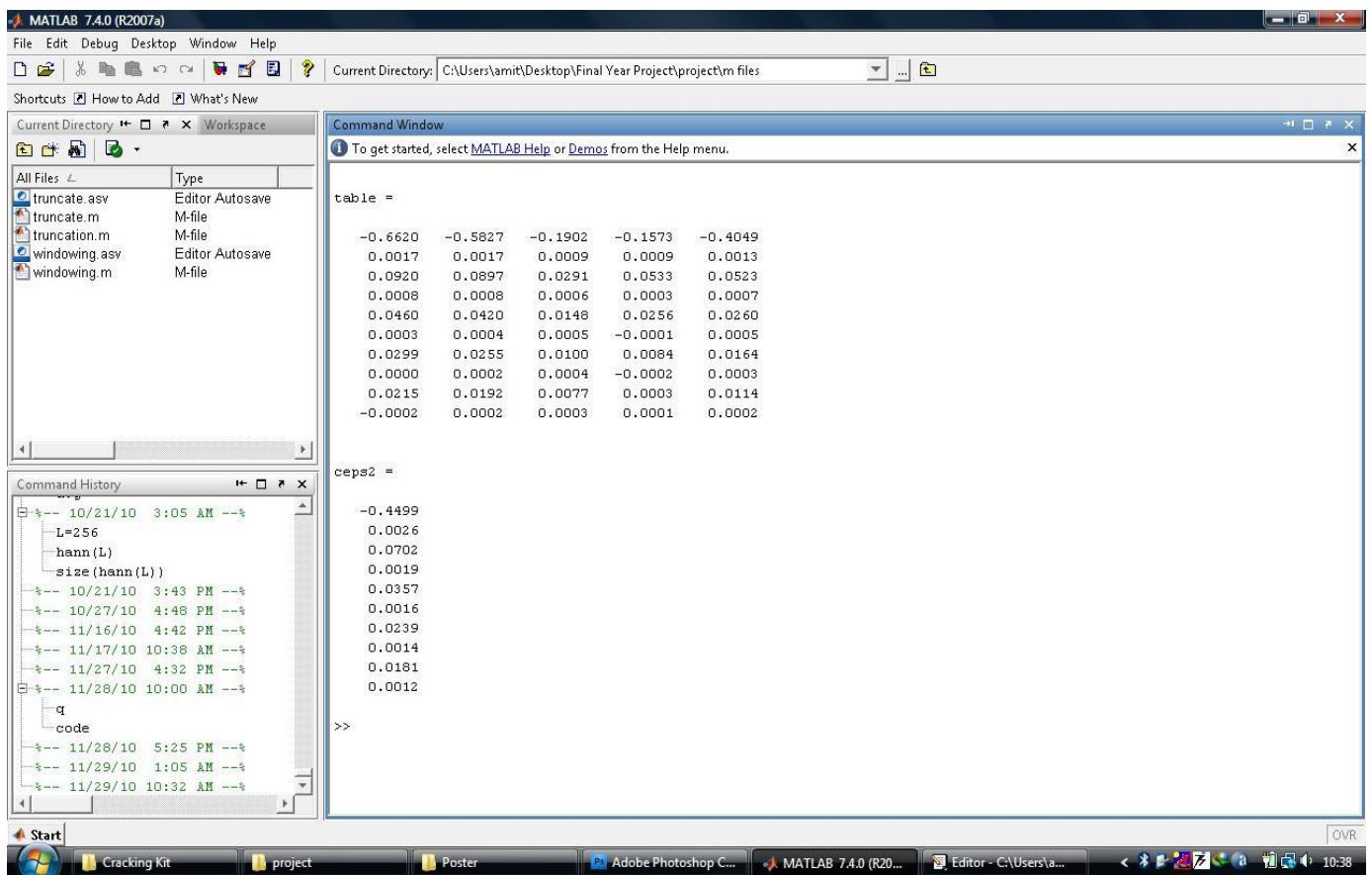
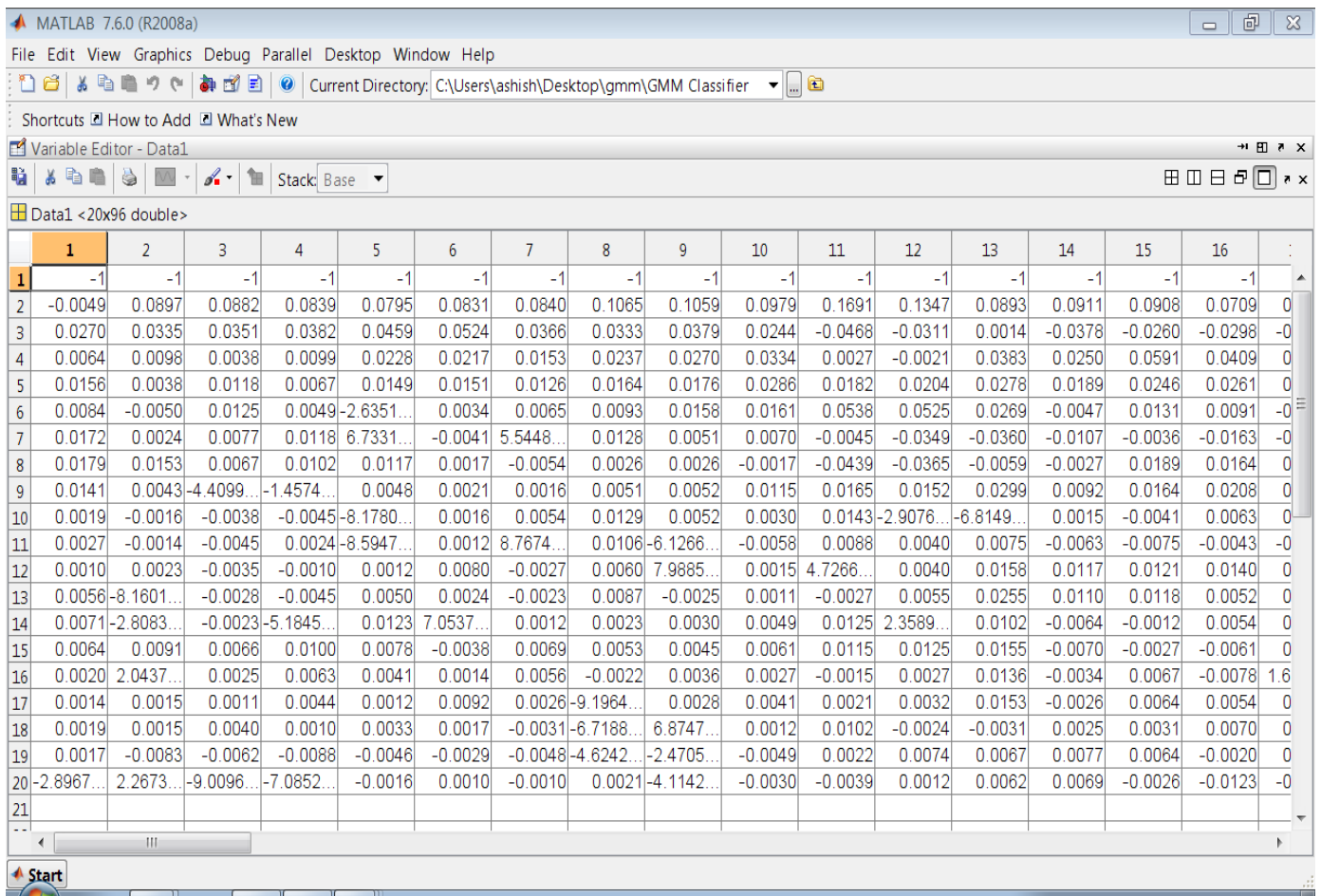


Figure 5.1: Result of Cepstral Coefficient Calculation

Above figure was obtained after calculating the Cepstral coefficients in MATLAB for five users each having five utterances of the word “hello”. Then it was averaged and represented in tabular form named “table”. Each column corresponds to a given speaker. The next column denoted as “ceps2” is Cepstral coefficient of 2nd speaker. We can clearly see its resemblance to 2nd column of “table”.

5.2.2 MFCC

It takes into account physiological behavior of perception of human ear which follows linear scale up to 1000 Hz and then follows log scale. Hence we convert frequency to mel domain using a number of filters. Then we take its absolute value, apply log function and convert back them into time domain using dct. For each user we had feature vectors having 20 MFCC coefficients each. For visualization purpose we only show few feature vectors and their MFCC.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	...
2	-0.0049	0.0897	0.0882	0.0839	0.0795	0.0831	0.0840	0.1065	0.1059	0.0979	0.1691	0.1347	0.0893	0.0911	0.0908	0.0709	0
3	0.0270	0.0335	0.0351	0.0382	0.0459	0.0524	0.0366	0.0333	0.0379	0.0244	-0.0468	-0.0311	0.0014	-0.0378	-0.0260	-0.0298	-0
4	0.0064	0.0098	0.0038	0.0099	0.0228	0.0217	0.0153	0.0237	0.0270	0.0334	0.0027	-0.0021	0.0383	0.0250	0.0591	0.0409	0
5	0.0156	0.0038	0.0118	0.0067	0.0149	0.0151	0.0126	0.0164	0.0176	0.0286	0.0182	0.0204	0.0278	0.0189	0.0246	0.0261	0
6	0.0084	-0.0050	0.0125	0.0049	-2.6351...	0.0034	0.0065	0.0093	0.0158	0.0161	0.0538	0.0525	0.0269	-0.0047	0.0131	0.0091	-0
7	0.0172	0.0024	0.0077	0.0118	6.7331...	-0.0041	5.5448...	0.0128	0.0051	0.0070	-0.0045	-0.0349	-0.0360	-0.0107	-0.0036	-0.0163	-0
8	0.0179	0.0153	0.0067	0.0102	0.0117	0.0017	-0.0054	0.0026	0.0026	-0.0017	-0.0439	-0.0365	-0.0059	-0.0027	0.0189	0.0164	0
9	0.0141	0.0043	-4.4099...	-1.4574...	0.0048	0.0021	0.0016	0.0051	0.0052	0.0115	0.0165	0.0152	0.0299	0.0092	0.0164	0.0208	0
10	0.0019	-0.0016	-0.0038	-0.0045	-8.1780...	0.0016	0.0054	0.0129	0.0052	0.0030	0.0143	-2.9076...	-6.8149...	0.0015	-0.0041	0.0063	0
11	0.0027	-0.0014	-0.0045	0.0024	-8.5947...	0.0012	8.7674...	0.0106	-6.1266...	-0.0058	0.0088	0.0040	0.0075	-0.0063	-0.0075	-0.0043	-0
12	0.0010	0.0023	-0.0035	-0.0010	0.0012	0.0080	-0.0027	0.0060	7.9885...	0.0015	4.7266...	0.0040	0.0158	0.0117	0.0121	0.0140	0
13	0.0056	-8.1601...	-0.0028	-0.0045	0.0050	0.0024	-0.0023	0.0087	-0.0025	0.0011	-0.0027	0.0055	0.0255	0.0110	0.0118	0.0052	0
14	0.0071	-2.8083...	-0.0023	-5.1845...	0.0123	7.0537...	0.0012	0.0023	0.0030	0.0049	0.0125	2.3589...	0.0102	-0.0064	-0.0012	0.0054	0
15	0.0064	0.0091	0.0066	0.0100	0.0078	-0.0038	0.0069	0.0053	0.0045	0.0061	0.0115	0.0125	0.0155	-0.0070	-0.0027	-0.0061	0
16	0.0020	2.0437...	0.0025	0.0063	0.0041	0.0014	0.0056	-0.0022	0.0036	0.0027	-0.0015	0.0027	0.0136	-0.0034	0.0067	-0.0078	1.6
17	0.0014	0.0015	0.0011	0.0044	0.0012	0.0092	0.0026	-9.1964...	0.0028	0.0041	0.0021	0.0032	0.0153	-0.0026	0.0064	0.0054	0
18	0.0019	0.0015	0.0040	0.0010	0.0033	0.0017	-0.0031	-6.7188...	6.8747...	0.0012	0.0102	-0.0024	-0.0031	0.0025	0.0031	0.0070	0
19	0.0017	-0.0083	-0.0062	-0.0088	-0.0046	-0.0029	-0.0048	-4.6242...	-2.4705...	-0.0049	0.0022	0.0074	0.0067	0.0077	0.0064	-0.0020	0
20	-2.8967...	2.2673...	-9.0096...	-7.0852...	-0.0016	0.0010	-0.0010	0.0021	-4.1142...	-0.0030	-0.0039	0.0012	0.0062	0.0069	-0.0026	-0.0123	-0
21																	

Figure 5.2: Feature vectors and MFCC

In the above figure we have only shown few feature vectors. Each column refers to a feature vector. The elements of each column are the corresponding MFCCs. As we had chosen first 20 DCT coefficients, hence each column will be having 20 elements.

5.3 FEATURE MATCHING

5.3.1 VQ USING LBG ALGORITHM

In VQ we went for efficient representation of feature vectors in feature space. Instead of handling large number of vectors we can cluster them to “Voronoi” region and each region can be represented by centroid called code word. Collection of all code words generates a code book which acts as a template for a given speaker. When an unknown utterance is encountered, VQ distortion for each feature vector is compute and summed up. A MATLAB program has been written for LBG algorithm and the subsequent figure illustrates this.

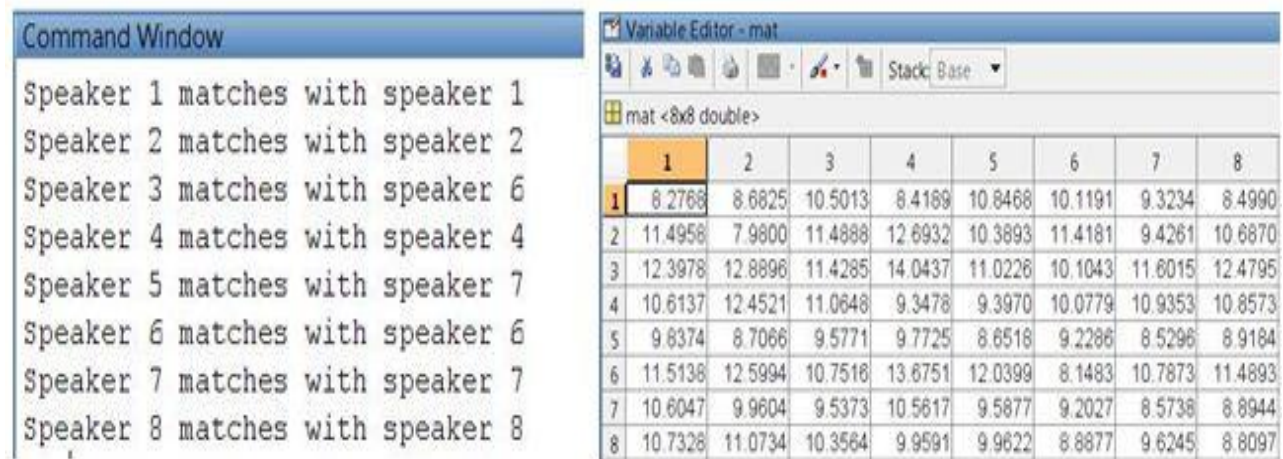


Figure 5.3: VQLBG output and corresponding VQ distortion matrix

As we can clearly infer while speakers 1, 2, 4, 6, 7 & 8 were correctly identified rest weren't. The matrix adjacent to “command window” displays the VQ distortion for all speakers. Each row & column refers to test speaker and train speaker respectively. Each element of a single row is the total VQ distortion obtained by using the code book of corresponding train speaker. To illustrate the process we can just focus on row1. Here 1st element has the least value. Hence the algorithm assigns speaker 1 as identity for 1st row unknown speaker. This is same for all unknown speakers. The least value in the row gives the identity of test speaker. Accuracy obtained was 75%.

5.3.2 DTW (DYNAMIC TIME WARPING) ALGORITHM

In our project work we have only implemented classical DTW method. In this method we first computed the Euclidian matrix of test utterance and unknown speaker. Then we tried to find out the optimal path giving least Euclidian sum using dynamic programming. We computed a cumulative distance matrix and its highest location gave the total Euclidian sum for optimal path. This process was repeated for all speakers in reference and a matrix d was computed using those values. The least value gave the desired user. The following figure best illustrates the above procedure:-

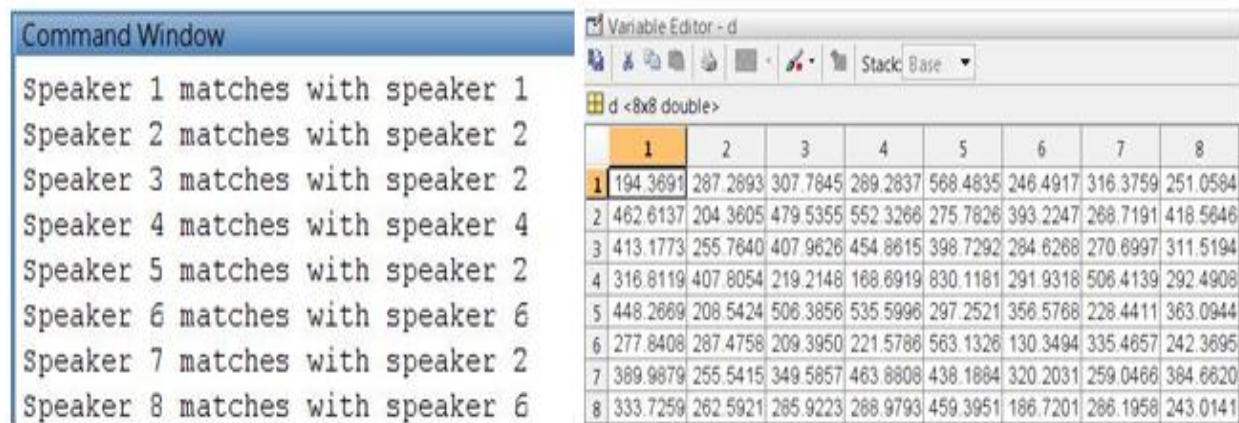
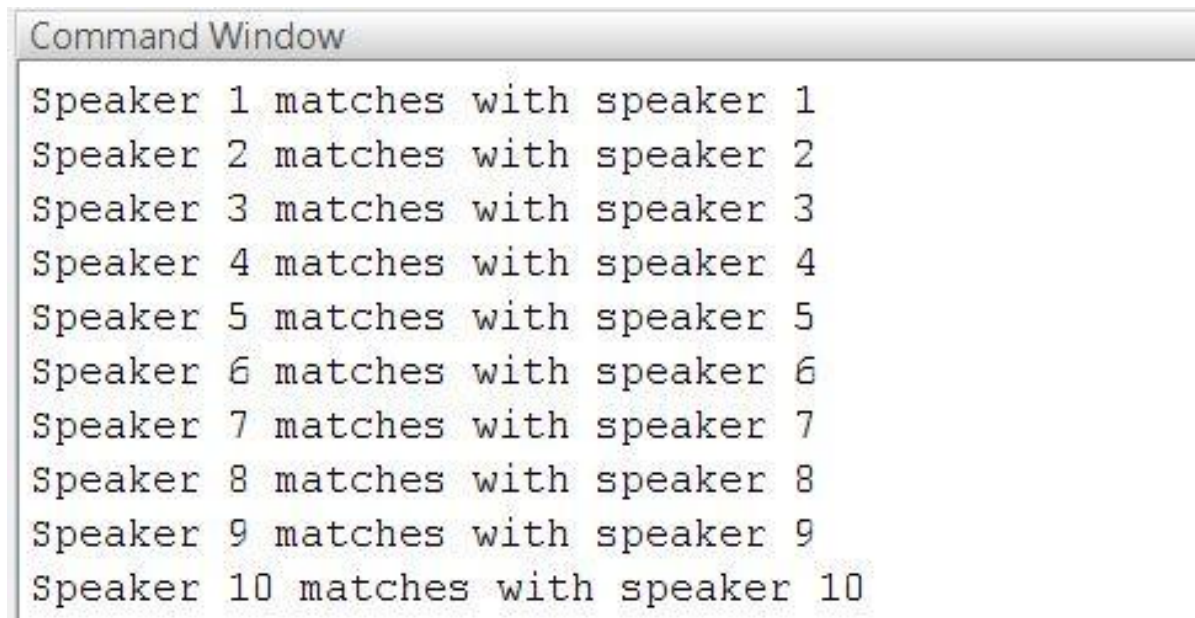


Figure 5.4: Speaker recognition using DTW

As we can clearly infer while speakers 1, 2, 4 & 6 were correctly identified rest weren't. The matrix adjacent to "command window" displays the Euclidian sum of optimal path for all speakers. Each row & column refers to test speaker and train speaker respectively. Each element of a single row is summation of Euclidian distance for optimal match between unknown speaker 1 and reference speaker of corresponding column. To illustrate the process we can just focus on row1. Here 1st element has the least value. Hence the algorithm assigns speaker 1 as identity for 1st row unknown speaker because out of all Euclidian sum this value is the least. This is same for all unknown speakers. The least value in the row gives the identity of test speaker. Accuracy obtained was 50%.

5.3.3 GMM (GAUSSIAN MIXTURE MODELING)

GMM assumes vector space to be divided into specific components depending on clustering of feature vectors and frames the feature vector distribution in each component to be Gaussian. As initially we have no idea about which vector belongs to which component a likelihood maximization algorithm is followed for optimal classification. For testing purpose we calculated posteriori probability of test utterance and the reference speaker maximizing Gaussian distribution is termed as identity of unknown speaker. The output of this procedure is given below:-



```
Command Window
Speaker 1 matches with speaker 1
Speaker 2 matches with speaker 2
Speaker 3 matches with speaker 3
Speaker 4 matches with speaker 4
Speaker 5 matches with speaker 5
Speaker 6 matches with speaker 6
Speaker 7 matches with speaker 7
Speaker 8 matches with speaker 8
Speaker 9 matches with speaker 9
Speaker 10 matches with speaker 10
```

Figure 5.5: Output of GMM

The accuracy obtained using this algorithm was 100%.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The results obtained using MFCC and VQ are appreciable. MFCCs for each speaker were computed and vector quantized for efficient representation. The code books were generated using LBG algorithm which optimizes the quantization process. VQ distortion between the resultant codebook and MFCCs of an unknown speaker was taken as the basis for determining the speaker's authenticity. Accuracy of 75% was obtained using VQLBG algorithm. It can be optimized by using high quality audio devices in a noise free environment. Use of more number of centroids increases the performance factor but degrades the computational efficiency. Hence an economical trade-off between code vectors and number of computation is required for optimized performance of VQLBG algorithm.

The next method implemented was DTW. It has its own virtues of being very simple and astonishingly computation efficient. Instead of data sample, MFCCs of a test utterance were warped with respect to reference speaker and the least Euclidian distance was taken as basis for speaker identification. Accuracy obtained using this method was 50%. This is because DTW doesn't take into account vocal tract information of a particular user. It only tries to align two vectors efficiently in time domain. Still its simplicity and easy hardware implementation has made it a regular tool for mobile applications.

Then we went for another method for speaker identification known as GMM. GMM models are motivated by the facts that vocal tract information of a speaker follows Gaussian distribution and Gaussian model approximates the feature space as a smooth surface. Accuracy obtained using GMM for same data set was 100% which clearly indicates its high efficiency. When number of components used becomes high its computational efficiency degrades a bit. But when we go for its high accuracy, these lacunas can be compromised.

REFERENCES

- [1] Campbell, J.P., Jr.; “Speaker recognition: a tutorial” Proceedings of the IEEE Volume 85, Issue 9, Sept. 1997 Page(s):1437 – 1462.

- [2] Seddik, H.; Rahmouni, A.; Samadhi, M.; “Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier” First International Symposium on Control, Communications and Signal Processing, Proceedings of IEEE 2004 Page(s):631 – 634.

- [3] Childers, D.G.; Skinner, D.P.; Kemerait, R.C.; “The cepstrum: A guide to processing” Proceedings of the IEEE Volume 65, Issue 10, Oct. 1977 Page(s):1428 – 1443.

- [4] Roucos, S. Berouti, M. Bolt, Beranek and Newman, Inc., Cambridge, MA; “The application of probability density estimation to text-independent speaker identification” IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '82. Volume: 7, On page(s): 1649- 1652. Publication Date: May 1982.

- [5] Castellano, P.J.; Slomka, S.; Sridharan, S.; “Telephone based speaker recognition using multiple binary classifier and Gaussian mixture models” IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 Volume 2, Page(s) :1075 – 1078 April 1997.

- [6] Zilovic, M.S.; Ramachandran, R.P.; Mammone, R.J “Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions”.; IEEE Transactions on Speech and Audio Processing, Volume 6, May 1998 Page(s):260 – 267

- [7] Davis, S.; Mermelstein, P, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences” , IEEE Transactions on Acoustics, Speech, and Signal Processing Volume 28, Issue 4, Aug 1980 Page(s):357 – 366

- [8] Y. Linde, A. Buzo & R. Gray, “An algorithm for vector quantizer design”, IEEE Transactions on Communications, Vol. 28, issue 1, Jan 1980 pp.84-95.
- [9] S. Furui, “Speaker independent isolated word recognition using dynamic features of speech spectrum”, IEEE Transactions on Acoustic, Speech, Signal Processing, Vol.34, issue 1, Feb 1986, pp. 52-59.
- [10] Fu Zhonghua; Zhao Rongchun; “An overview of modeling technology of speaker recognition”, IEEE Proceedings of the International Conference on Neural Networks and Signal Processing Volume 2, Page(s):887 – 891, Dec. 2003.
- [11] PRADEEP. CH,” TEXT DEPENDENT SPEAKER RECOGNITION USING MFCC AND LBG VQ”, National Institute of Technology, Rourkela, 2007
- [12] Eamonn J. Keogh[†] and Michael J. Pazzani[‡] ,” Derivative Dynamic Time Warping”, Department of Information and Computer Science University of California, Irvine, California 92697 USA
- [13] Douglas A. Reynolds and Richard C. Rose, “Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models“, IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 3, NO. 1, JANUARY 1995
- [14] A. Dempster, N. Laird, and D. Rubin, “Maximum Likelihood from incomplete data via the EM algorithm, ” J.Royal Stat. Soc., vol 39, pp. 1-38, 1977.
- [15] Reynolds D.A.: “A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification”, Ph.D. thesis, Georgia Institute of Technology, September 1992.
- [16] David P., Nouza J.: “Úloha rozpoznávání mluvčího”, “Počítačové zpracování řeči – cíle, problémy, metody a aplikace”, pp. 95-105, December 2001.
- [17] Linde Y., Buzo A., and Gray R. M.: “An Algorithm for Vector Quantizer Design”, IEEE Transactions on Communications, pp. 702-710, January 1980
- [18] www.wikipedia.org